What is claimed is:

1. An apparatus comprising:

a data input;

a data output;

a multiplexer have a plurality of inputs and an output coupled to said data output, and having a control input for receiving a switching control signal;

a comparator having a plurality of data inputs and having a reference bit pattern, and having one primitive present output corresponding to each data input, said comparator functioning to compare the data at each input to said reference bit pattern and activate a primitive present signal on the output corresponding to each of said data inputs at which data appears which matches said reference bit pattern;

a plurality of data registers having data inputs coupled to said data input, each said data register having a data output coupled to one of said inputs of said multiplexer, and to one of said inputs of said comparator, each said data register having an address input for receiving a select input signal, which when active, will cause said data register to store the data then existing on said data input;

a receive clock input for receiving a receive clock signal;

a transmit clock input for receiving a transmit clock signal;

an insertion/deletion control unit having a plurality of inputs coupled to receive said primitive present signals from said comparator, and having a transmit address input for receiving a transmit address signal and a receive address input for receiving a receive address signal, and having a switching control output coupled to supply said switching control signal to said control input of said multiplexer, and having a delete output at which a Delete signal appears, and having an insert output at which an Insert signal appears, and having logic for comparing a transmit address to said receive address and activating said Delete signal when the difference between said transmit address and said receive address is greater than the number of data registers divided by two, and for activating said Insert signal when the difference between said transmit address and said receive address is less than the number of data registers divided by two, said transmit address controlling which of said outputs of said data registers is coupled through said multiplexer to said data output, and said receive address controlling which of said data registers stores the data currently at

said data input;

a receive address counter having a clock input coupled to said receive clock signal and having an address output coupled to each of said address inputs of said plurality of data registers for generating said select input signals for said data registers in sequence as said receive clock increments, and having receive address output coupled to said receive address input of said insertion/deletion control unit to supply said receive address signal thereto;

a transmit address counter having a clock input coupled to receive said transmit clock signal, and having a transmit address output coupled to said transmit address input of said insertion/deletion logic to supply said transmit address signal to said insertion/deletion logic for supplying by said insertion/deletion logic to said multiplexer as said switching control signal, and having an input for receiving a Delete signal and an input for receiving an Insert signal, said transmit address counter configured skip the address in the sequence of address incrementations that corresponds to the address of the data register in which a deletable primitive is stored when said delete signal is activated and configured to dwell on the address that corresponds to the address of the data register in which a deletable non essential primitive or other non essential data is stored for at least one extra clock cycle of said transmit clock when said Insert signal is activated.

2. An apparatus comprising:

means for receiving at a receive clock rate a stream of serial format data including data words and primitives which can be deleted without adverse effects, hereafter called nonessential primitives, and/or other nonessential data and storing said received data in a first in, first out buffer, each data word, primitive and piece of nonessential data stored at a different address;

means for transmitting at a transmit clock rate the data words, primitives and nonessential data stored in said FIFO at selected addresses;

means for determining which addresses in said FIFO store primitives or nonessential data that can be deleted;

means for comparing a transmit address pointer in said FIFO which is incremented at said transmit clock rate to a receive address pointer in said FIFO

which is incremented at said receive clock rate and, when the distance between said pointers indicates the possibility of overflow or underflow, controlling said means for transmitting so as to insert nonessential primitives or other nonessential data or to delete nonessential primitives or other nonessential data appropriately to avert either overflow or underflow.

3. A process for preventing overflow or underflow in serial data transmission protocols with separate transmit and receive clocks which may be running at different frequencies, comprising the steps of:

receiving at a receive clock rate a stream of serial format data including data words and primitives which can be deleted without adverse effects, hereafter called nonessential primitives, and/or other nonessential data and storing said received data in a first in, first out buffer, each data word, primitive and piece of nonessential data stored at a different address;

transmitting at a transmit clock rate the data words, primitives and nonessential data stored in said FIFO at selected addresses;

determining which addresses in said FIFO store primitives or nonessential data that can be deleted;

comparing a transmit address pointer in said FIFO which is incremented at said transmit clock rate to a receive address pointer in said FIFO which is incremented at said receive clock rate and, when the distance between said pointers indicates the possibility of overflow or underflow, controlling said means for transmitting so as to insert nonessential primitives or other nonessential data or to delete nonessential primitives or other nonessential data appropriately to avert either overflow or underflow.